Package 'ssM8'

24 September 2010

Version 2.1-8

Date 2010-09-24

Title M8 Earthquake Forecasting Algorithm

Author David Harte, ported to R and packaged by Ray Brownrigg

Maintainer David Harte <david@statsresearch.co.nz>

Description This package performs the M8 calculations as in Kossobokov (1997). It was originally written for S-PLUS, being part of the Statistical Seismology Library (Harte, 1998), and was translated to R by Ray Brownrigg <Ray.Brownrigg@vuw.ac.nz>. More information about the M8 Algorithm can be found under the topic "changes" along with recent modifications made to the package.

Depends ssBase

Suggests ssNZ

License GPL (>=2)

URL http://www.statsresearch.co.nz/software.html

R topics documented:

Change Log 2
decluster.M8 4
emp.cdf 5
M8 7
M8.series 8
M8.TIP 11
plot.M8 14

Index **16**

`Change Log` *Changes Made to the Package*

Description

The M8 algorithm is based on seven different time series (see Keilis-Borok \& Kossobokov, 1990; and Kossobokov, 1997). If the 7th time series and 5 out of the first 6 time series are above certain thresholds, a "Time of Increased Probability" (or TIP) is declared. This means there is an increased probability of an earthquake (of or greater than M_0) in the next 5 years. Details for the calculation of the seven series can be found in under `M8.series`, and details for the calling of a TIP can be found under `M8.TIP`.

Li (1997) converted the original M8 FORTRAN code in Kossobokov (1997) into S-PLUS, which was then included into the Statistical Seismology Library (Harte, 1998). Ray Brownrigg subsequently packaged it into R.

Analyses using the M8 Algorithm, applied to New Zealand data, have been carried out by Ma \& Vere-Jones (1997), Harte et al. (2003), and Harte et al. (2007).

This page contains a listing of recent changes made to functions, and known general problems.

Details

1. `decluster.M8`: The line near the bottom within an else loop: `class(minday) <- "datetimes"` has been commented. (March 2002)
2. `plot.M8`: Main title now uses default `cex.main`, vertical lines (`type="h"`) have been removed, horizontal reference line changed to dashed (`lty=2`), colours removed from plotted symbols, and 8th graph plots only critical series. (January 2003)
3. Occurrences of `T` and `F` have been changed to `TRUE` and `FALSE`, respectively. (January 2003)
4. `M8.TIP`: When `(training == "moving")` the line `TIP.type[1:7] <- NA` has been changed to `TIP.type[1:7] <- ""`. When `(training == "user" || training == "all")` the line `TIP.type[1] <- NA` has been changed to `TIP.type[1] <- ""`. The earlier versions caused errors when running `R CMD check` in R\ version 1.6.2. (February 2003)
5. `plot.M8`: argument `M8` changed to `x`, and argument ... added to fix generic/method inconsistencies. (June 2003)
6. `print.mainshocks` has been renamed to `prt.mainshocks` to fix generic/method confusion. (June 2003)
7. `time.grid` has been renamed to `timegrid` to fix generic/method confusion. (June 2003)
8. `M8.TIP`: the check to see if the critical series has length less than seven now uses function `nrow` rather than `start` and `end`. (October 2003)
9. `decluster.M8`: a vector is now returned that identifies the mainshock to which each aftershock belongs. (October 2003)
10. Minor formatting corrections to manual pages. (December 2003)
11. Minor documentation formatting changes, mainly to use `\dQuote`. (January 2004)
12. `in.circle`: correct documentation mismatch with code. (7 May 2004)
13. `decluster.M8`: `PACKAGE="ssM8"` has been added to the `.Fortran("recat", ...)` call within the function. (7 May 2004)
14. `sum.row` renamed to `sumrow`. (7 May 2004)

15. A statement `require(ssNZ)` has been added to the examples that use the NZ catalogue. Hence **ssNZ** is no longer a "package" requirement of **ssM8**. (25 May 2004)
16. `M8.series`: statement with incorrect logical syntax corrected:

    ```
    cover.breaks <-

    (time.breaks[1] <= catalogue[, "time"] <= time.breaks[n])
    ```

    (17 Feb 2005)
17. `M8.TIP`: make documentation consistent with Harte et al (2003). (17 Feb 2005)
18. `M8.TIP`: method of calculating the quantiles in variable `tops` incorrect, redone using the function `quantile`. Values of `g` and `h` included in output list object. Variable initialisation rewritten. Code included to calculate `TIP.level` when `training=="moving"`, previously not working for this option. (24 Mar 2005)
19. `emp.cdf`: completely rewritten, see documentation. (11 Apr 2005)
20. Old package "stepfun" not now "required". Part of **base**. (19 Apr 2005)
21. `decluster.M8`: on manual page, units added for space and time windows. (9 Jul 2006)
22. All occurences of `subset.rect` in the Examples have been changed to `subsetrect`. See Changes in **ssBase**. (8 Nov 2007)
23. **ssNZ** added to `Suggests` in the DESCRIPTION file. Corresponding changes made to examples on manual pages. (10 Jan 2008)
24. Use of `require(ssNZ)` in examples made consistent with package **ssEDA**. (28 Jan 2008)
25. References updated below. (28 Jan 2008)
26. Package vignettes added. (28 Jan 2008)
27. Removal of LaTeX markups from DESCRIPTION file. (31 May 2009)
28. Minor manual page changes to conform to R\ 2.10.0. (13 Nov 2009)
29. `M8.TIP`: called a function `stoptmp`, non-existent, changed to `stop`. (13 Nov 2009)
30. `timegrid`: internal function, not used, deleted. (17 Nov 2009)
31. `Changes`: correct hyperlink from package vignettes. (11 Feb 2010)
32. Add CITATION file. (24 Sep 2010)

References

Harte, D. (1998). Documentation for the Statistical Seismology Library. School of Mathematical and Computing Sciences Research Report No. 98-10 (Updated Edition June 1999), Victoria University of Wellington. (ISSN 1174-4545)

Harte, D.; Li, D.-F.; Vreede, M. \& Vere-Jones, D. (2003). Quantifying the M8 prediction algorithm: Reduction to a single critical variable and stability results. *NZ Journal of Geology and Geophysics* **46(1)**, 141–152. URL: http://www.rsnz.org/publish/nzjgg/2003/010.php

Harte, D.; Li, D.-F.; Vere-Jones, D.; Vreede, M. \& Wang, Q. (2007). Quantifying the M8 algorithm: Model, forecast and evaluation. *NZ Journal of Geology and Geophysics* **50(2)**, 117–130. URL: http://www.rsnz.org/publish/nzjgg/2007/012.php

Keilis-Borok, V.I. and Kossobokov, V.G. (1990). Premonitory activation of earthquake flow: algorithm M8. *Physics of the Earth and Planetary Interiors* **61**, 73–83. URL: http://dx.doi.org/10.1016/0031-9201(90)90096-G

Kossobokov, V.G. (1997). User manual for M8. In: *Algorithms for Earthquake Statistics and Prediction. International Association of Seismology and Physics of the Earth's Interior (IASPEI)*

Library Volume 6. Pages 167–222. (Edited by J.H. Healy, V.I. Keilis-Borok and W.H.K. Lee). IASPEI, Menlo Park CA.

Li, Dongfeng. (1997). M8-Splus Documentation. Internal Report, Institute of Statistics and Operations Research, Victoria University of Wellington.

Ma, L. and Vere-Jones, D. (1997). Application of the M8 and Lin-Lin algorithms to New Zealand Earthquake data. *NZ Journal of Geology and Geophysics* **40**, 77–89. URL: http://www.rsnz.org/publish/nzjgg/1997/7.php

---

`decluster.M8` *Decluster Catalogue Using M8 Method*

Description

The function reads an earthquake catalogue (with class `"catalogue"`), deletes events with magnitude below a cutoff level and all aftershocks, optionally deletes events with depth too deep or too shallow, and counts the number of aftershocks in the 14 days following each remaining mainshock. The output is also a `"catalogue"` object containing the mainshock information and an additional variable with the corresponding number of aftershocks.

Usage

```
decluster.M8(catalogue, cutoff.mag=4, minday=min(catalogue$time),
mindepth= - Inf, maxdepth=Inf, debug=FALSE, decluster.name="",
magn.window = c(3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8),
space.window = c(30, 40, 40, 40, 50, 50, 50, 100, 100, 150, 200),
time.window = c(6, 11, 23, 46, 91, 183, 183, 365, 730, 913, 1096))
```

Arguments

`catalogue`	the catalogue. It should be a `"catalogue"` object.
`decluster.name`	character string containing the name of the declustered catalogue.
`cutoff.mag`	the magnitude below which the record is cutoff before going through the mainshock-aftershock separation. Default 4.0.
`minday`	number of days from 1 January 1970 (julian). If specified, then after mainshocks have been separated out, mainshocks before this date are dropped. If not specified, no mainshocks are dropped.
`mindepth`	all events with depth less than mindepth are deleted before entering algorithm. Default is `-Inf`.
`maxdepth`	all events with depth greater than or equal to maxdepth are deleted before entering algorithm. Default is `Inf`.
`magn.window`	a vector giving the magnitude window boundaries. The default `c(3, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0)`.
`time.window`	a vector giving the time window boundaries (days). The default is `c(6, 11, 23, 46, 91, 183, 183, 365, 730, 913, 1096)`.
`space.window`	a vector giving the space window boundaries (km). The default is `c(30, 40, 40, 40, 50, 50, 50, 100, 100, 150, 200)`.
`debug`	boolean; default is `FALSE`.

Details

All events with magnitude less than `magn.window[1]` are assumed to have no aftershocks. Those events with magnitude less than `magn.window[i+1]` but greater than or equal to `magn.window[i]` have aftershock time and space windows given by `time.window[i]` and `space.window[i]` respectively, for `i = 1, ..., length(magn.window)-1`. All events with magnitude greater than or equal to `magn.window[length( magn.window)]` have aftershock time and space windows given by `time.window[length(magn.window)]` and `space.window[length(magn.window)]` respectively.

Value

The value returned is the vector identifying the mainshock to which each aftershock belongs. The mainshocks themselves are identified by zeros.

A `"catalogue"` object with name given by the argument `decluster.name` is written to the session workspace. It contains the mainshock hypocentral information and an additional variable, `n.aftershocks`, containing the number of aftershocks following the mainshock within 14 days.

Author(s)

Li Dongfeng, 1997; modified November 1999

See Also

`M8`, `M8.series`, `M8.TIP`, `plot.M8`

Examples

```
#   Requires the package ssNZ
require(ssNZ)

as.catalogue(subsetrect(NZ, minday=julian(1,1,1965),
                        maxday=julian(1,1,2000), minmag=4.5),
             catname="NZ45")

decluster.M8(NZ45, cutoff.mag=4.5, decluster.name="NZ.mainshocks")

print(summary(NZ.mainshocks))
```

---

emp.cdf *Empirical Cumulative Distribution Function*

Description

Calculates the empirical cumulative probability distribution function. This is the inverse function of `quantile`.

Usage

```
emp.cdf(x, q)
```

Arguments

x	is a vector containing the sample data.
q	is a vector of points where the function is to be evaluated.

Details

This function is an *exact* inverse of the quantile function when the points in q are contained in x. This is not the case for the function ecdf.

For example, say x contains 11 unique points. Then according to the quantile function, the minimum represents the zero-th quantile, and the remaining sorted values represent the 10%ile, 20%ile, 30%ile, up to 100%ile. The function emp.cdf has been defined to be consistent with this.

This consistency is important for the evaluation of the M8 algorithm. The TIP declaration within the function M8.TIP is calculated by two different methods, one using the quantile function, and the other based on the empirical cumulative probability distribution function. If they are not consistent then the time point of the TIP declaration can be different between the two methods.

Value

A vector of the same length as q, containing the cumulative probabilities.

Author(s)

David Harte, 2005

See Also

quantile, ecdf

Examples

```
n <- 10
x <- rep(0:n, 1)
print(quantile(x, probs=seq(0, 1, 1/n)))
print(emp.cdf(x, x))

n <- 12
x <- rep(0:n, 3)
print(quantile(x, probs=seq(0, 1, 1/n)))
print(emp.cdf(x, x))

n <- 12
x <- rnorm(n+1)
print(quantile(x, probs=seq(0, 1, 1/n)))
print(sort(x))
print(emp.cdf(x, sort(x)))
```

M8 *Run the M8 Algorithm*

Description

This function runs the M8 algorithm, i.e. calls the function `M8.series` to calculate the seven time series, calls the function `M8.TIP` to calculate the Times of Increased Probabilities, and finally, optionally calls the function `plot.M8` to plot all seven series plus an eighth critical series.

Usage

```
M8(catalogue, M0, centrelong=mean(catalogue[, "longitude"]),
   centrelat=mean(catalogue[, "latitude"]), radius=radius.M8default(M0),
   minday=catalogue[1, "time"],
   start.series=datetimes(year=(years1(minday) + 12), month=1, day=1,
   hour=0, minute=0), training="user", end.training=NA, time.breaks=Inf,
   running.total=12, smoother=6, TIP.length=10, plotit=TRUE,
   title="M8 Series and TIPs", debug=FALSE)
```

Arguments

`catalogue`	the mainshock catalogue (created by the `decluster.M8` function), with the number of aftershocks in following 14 days.
`M0`	numeric. The aim of the algorithm is to predict earthquakes with a magnitude greater than or equal to M_0.
`centrelong`	longitude of the centre of the circle of interest.
`centrelat`	latitude of the centre of the circle of interest.
`radius`	circle of interest radius. The default radius is $55.5\{\exp(M_0 - 5.6) + 1\}$.
`minday`	Julian date, events earlier than this are not used in analysis. By default, all the data are used.
`start.series`	Julian date of end point of the first bin from which the series are generated. Since the series are calculated using bins (defined by `time.breaks`), the `start.series` must coincide with an end point of one of the bins. The M8 default is six month bins.
`training`	has three possible modes: `"moving"`, `"user"`, and `"all"`. See Details in `M8.series` for further explanation.
`end.training`	Julian date for the end of the training period. This date should be the end point of a bin. This is only used when the training mode is `"user"`.
`time.breaks`	a vector of Julian dates that give the bin boundaries for calculating the series. The (M8) default bin length is 6 months. Care must be taken that the arguments `minday`, `start.series` and `end.training` coincide with a breakpoint in the vector `time.breaks`. The breaks should also be at regular intervals.
`running.total`	by default, the series are calculated with a six year running total, i.e.12 half years, thus the M8 default is 12.
`smoother`	integer, denoting the length of the smoothing window. The series are smoothed at a given point by taking their maximum value over previous values within this window. The default value is 6 (i.e. three years).

TIP.length	is the duration of the TIP or "Time of Increased Probability". The M8 default is 5 years (i.e. 10 time intervals), and is independent of M0.
plotit	logical. If set to TRUE then seven plots for each of the seven smoothed series are generated. There is also a plot of the largest earthquake in each 6 month interval, together with the critical series. The default is TRUE.
title	character string giving a title for the page of plots. The default is "M8 Series and TIPs".
debug	logical. The default is FALSE.

Details

See Details in M8.series for an explanation of how the M8 series are calculated, and M8.TIP for how the Times of Increased Probabilities are calculated.

Value

The same object as that returned by the function M8.TIP.

Author(s)

Li Dongfeng, 1997; modified by Maaike Vreede, 1998.

See Also

decluster.M8, M8.series, M8.TIP

Examples

```
#   Requires the package ssNZ
require(ssNZ)

as.catalogue(subsetrect(NZ, minday=julian(1,1,1965),
                        maxday=julian(1,1,2000), minmag=4.5),
             catname="NZ45")

decluster.M8(NZ45, cutoff.mag=4.5, decluster.name="NZ.mainshocks")

savpar <- par(no.readonly=TRUE)
temp <- M8(NZ.mainshocks, M0=7.0, minday=julian(y=1965, x=1, d=1),
           start.series=julian(y=1975, x=1, d=1), centrelong=176,
           centrelat=-39, end.training=julian(y=2000, x=1, d=1))
par(savpar)
```

M8.series *Calculates the Seven M8 Series*

Description

This functions calculates the seven time series of the M8 Algorithm.

Usage

```
M8.series(catalogue, M0, centrelong=mean(catalogue[, "longitude"]),
          centrelat=mean(catalogue[, "latitude"]),
          radius=radius.M8default(M0), minday=catalogue[1, "time"],
          start.series=datetimes(year=(years1(minday) + 12), month=1,
          day=1, hour=0, minute=0), training="user", end.training=NA,
          time.breaks=Inf, running.total=12, debug=FALSE)
```

Arguments

`catalogue`	the mainshock catalogue (created by the `decluster.M8` function), with the number of aftershocks in following 14 days.
`M0`	numeric. The aim of the algorithm is to predict earthquakes with a magnitude greater than or equal value to M_0.
`centrelong`	longitude of the centre of the circle of interest.
`centrelat`	latitude of the centre of the circle of interest.
`radius`	circle of interest radius. The default radius is $55.5\{\exp(M_0 - 5.6) + 1\}$.
`minday`	Julian date, events earlier than this are not used in analysis. By default, all the data are used.
`start.series`	Julian date of end point of the first bin from which the series are generated. Since the series are calculated using bins (defined by `time.breaks`), the `start.series` must coincide with an end point of one of the bins. The M8 default is six month bins.
`training`	has three possible modes: `"moving"`, `"user"`, and `"all"`. See Details in `M8.series` for further explanation.
`end.training`	Julian date for the end of the training period. This date should be the end point of a bin. This is only used when the training mode is `"user"`.
`time.breaks`	a vector of Julian dates that give the bin boundaries for calculating the series. The (M8) default bin length is 6 months. Care must be taken that the arguments `minday`, `start.series` and `end.training` coincide with a breakpoint in the vector `time.breaks`. The breaks should also be at regular intervals.
`running.total`	by default, the series are calculated with a six year running total, i.e.12 half years, thus the M8 default is 12.
`debug`	logical. The default is `FALSE`.

Details

Initially, a training period is defined. This period is used to estimate various magnitude cutoffs, described below. Three time points are specified in the arguments to this function: `minday` $\leq$ `start.series` $\leq$ `end.training`. Further, there are three possible training modes:

`"usr"` the training period starts at `start.series` and finishes at `end.training`.

`"moving"` the training period starts at `minday` and finishes on the date of the last event in the catalogue.

`"all"` the training period starts at `start.series` and finishes on the date of the last event in the catalogue.

Using the training period, the algorithm then determines the required magnitude cutoffs to produce the following subcatalogues: CAT20, CAT10, CAT20a, CAT10a, and CATMS. The subset CAT20 has smaller earthquakes cut out so that it has an average of 20 earthquakes per year. Similarly for CAT10, though with an average of 10 events per year. The subset CAT20a is the same as CAT20, except that earthquakes stronger than a half magnitude below the target magnitude are omitted. Similarly for CAT10a. The subset CATMS contains the events in the catalogue that have magnitudes between $(M_0 - 2)$ and $(M_0 - 0.2)$, where M_0 is the target magnitude.

Using the vector `time.breaks`, the algorithm computes, for each time interval, the value of the seven series as follows. Note that these series start being calculated from the date `start.series`.

Series 1 is the number of earthquakes in CAT20.

Series 2 is the number of earthquakes in CAT10.

Series 3 and 4 are derived from series 1 and 2 respectively. They are the difference between the number of earthquakes in the past 6 years and the number expected, based on all data 6 years before the date when the series start being calculated.

Series 5 and 6 have the same lower magnitude thresholds as the first 4 series, but earthquakes used in their calculation must also be a at least half a point less in magnitude than M_0 (the magnitude targeted for prediction). Series 5 uses CAT20a, and series 6 uses CAT10a. These series are the average magnitude multiplied by the 2/3 power of the number of events.

Series 7 is based on CATMS. It is the maximum number of aftershocks after any event in this magnitude range over the past year.

These are regular time series, with intervals of 6 months, and a 6 year running total incorporated into their calculation. Before they are used for prediction they are smoothed again by taking the maximum value over the past 3 years. Then the M8 decision rule is applied: if 5 out of the first 6 series are above their 90th percentile, and the 7th series is above its 75th percentile, a "TIP" is declared.

Value

The function produces a list object with the following components.

`series` is a regular time series (rts) object with n rows and 8 columns. The first 7 columns are the seven series, and the last is the datetime corresponding to the series values.

`max.events` is a data frame with the same column structure as the catalogue. However, the columns are regular time series (with 6 month intervals) containing attributes of the largest earthquake in that 6 month period.

Author(s)

Li Dongfeng, 1997; modified by Maaike Vreede, 1998.

See Also

`decluster.M8`, `M8`, `M8.TIP`

Examples

```
#   Requires the package ssNZ
require(ssNZ)

as.catalogue(subsetrect(NZ, minday=julian(1,1,1965),
                        maxday=julian(1,1,2000), minmag=4.5),
             catname="NZ45")

decluster.M8(NZ45, cutoff.mag=4.5, decluster.name="NZ.mainshocks")

temp <- M8.series(NZ.mainshocks, M0=7.0, minday=julian(y=1965, x=1, d=1),
                  start.series=julian(y=1975, x=1, d=1), centrelong=176,
                  centrelat=-39, end.training=julian(y=2000, x=1, d=1))

print(temp$series)
```

`M8.TIP` *M8 Times of Increased Probability*

Description

Calculates the times of increased probability for the M8 Algorithm.

Usage

```
M8.TIP(series, M0, training="user", end.training=NA, smoother=6,
       TIP.length=10, debug=FALSE)
```

Arguments

`series`	a list object produced by the function `M8.series`.
`M0`	numeric. The aim of the algorithm is to predict earthquakes with a magnitude greater than or equal value to this value.
`training`	has three possible modes: `"moving"`, `"user"`, and `"all"`. See Details in `M8.series` for further explanation.
`end.training`	Julian date for the end of the training period. This date should be the end point of a bin. This is only used when the training mode is `"user"`.
`smoother`	integer, denoting the length of the smoothing window. The series are smoothed at a given point by taking their maximum value over previous values within this window. The default value is 6 (i.e. three years).
`TIP.length`	is the duration of the TIP or "Time of Increased Probability". The M8 default is 5 years (i.e. 10 time intervals), and is independent of `M0`.
`debug`	logical. The default is `FALSE`.

Details

The empirical distribution of each series is initially calculated. The way this is done depends on the training mode. If the training mode is `"moving"`, only the threshold percentiles for declaring a TIP are calculated. This is because the `"moving"` mode emulates the position of someone making real-time predictions - the empirical percentiles are updated with each new data point. Before the empirical series are used for prediction they are smoothed by taking the maximum value over the past 3 years (argument `smoother` = 6).

If the training mode is `"all"`, the empirical distribution of the series is calculated using all available values of the series. If the mode is `"user"`, only values of series during the training period are used to calculate the empirical distribution. The `rts` objects `tops` and `exceeds` are computed. The object `tops` gives the threshold percentiles for each series relevant at time i, and `exceeds` is a logical `rts` indicating whether the series exceeds the threshold percentile in the time interval J_i.

`M8.TIP` then looks at the decision rule for declaring a TIP. For a TIP to be declared, 5 out of the first 6 series must cross their 90th percentile (at least once in the last 3 years), and the 7th series must also cross its 75th percentile (also at least once in the last 3 years). This was formalised by Harte et al (2003) as follows. Denote the mth series in interval J_i as $F_m(J_i)$.

1. Let
$$U_m(J_i) = G_m[F_m(J_i)]\,,$$
where G_m is the empirical distribution function of the mth series. Then $100\,U_m(J_i)$ is the percentile value corresponding to the series value $F_m(J_i)$.
2. Find the maximum for each in the preceding 3 years, i.e.
$$V_m(J_i) = \max\left\{U_m(J_i), U_m(J_{i-1}), \cdots, U_m(J_{t-5})\right\}.$$
3. Now let $V_{[2]}(J_i)$ denote the second smallest of the six values $V_1(J_i), \cdots, V_6(J_i)$. Then the `TIP.level` is defined as
$$W(J_i) = \min\left\{V_{[2]}(J_i) - 0.9, V_7(J_i) - 0.75\right\}.$$
We refer to $W(J_i)$ as the *critical series* (see Harte et al, 2003). Values are listed in the output object as `TIP.level`.

A TIP is declared if $W(J_i) \geq 0$ for *two consecutive intervals*. Hence, by letting
$$Y(J_i) = \min\{W(J_{i-1}), W(J_i)\}\,,$$
a TIP is declared in time interval J_i if $Y(J_i) \geq 0$, and then the value of `TIP` in the output object is `TRUE`, otherwise `FALSE`. The duration of a TIP is specified by the argument `TIP.length`, which is 5 years by default.

If an earthquake with magnitude greater than or equal to the target magnitude occurs, the character variable `TIP.type` is set to `"STIP"` (successful TIP). If an earthquake with magnitude smaller than M_0, but $\geq M_0 - 0.5$ occurs, `TIP.type` is set to `"STIP-"` (nearly successful TIP). If the TIP is triggered by an earthquake with magnitude greater than or equal to the target magnitude, `TIP.type` is set to `"c.e."`. If no earthquake with magnitude $\geq M_0$ occurs, `TIP.type` is set to `"FTIP"` (false TIP), and if the 5 year duration of the TIP has not come to an end (and no earthquake $\geq M_0$ has occurred) `TIP.type` is set to `"CTIP"` (continuing TIP).

Value

A list object with the same components as that returned by `M8.series`, but with the additional variables:

`tops` is a rts with n rows and 7 columns. It contains the historical top 10 percent (for the first 6 series) or 25 percent of values (for the 7th series) for each 6 month interval. The percentile depends on the training mode.

`exceeds` is a rts with n rows and 7 columns. It is logical, indicating whether each series exceeds the relevant historical percentiles for each six month interval.

`TIP` is a logical variable indicating whether a TIP is declared in each 6 month interval, see $Y(J_i)$ in "Details".

`TIP.type` classifies the TIPS with the following character strings:

- `"c.e."` earthquake caused, that is, large earthquake $\geq M_0$ in the preceding year of the TIP declaration.
- `"STIP"` successful warning, large event occurred in the 5 year period after declaration.
- `"FTIP"` failed warning, large event did not occur in the 5 year period after declaration.
- `"CTIP"` current warning, not 5 years past since declaration.
- `"STIP-"` between STIP and FTIP, strictly speaking is FTIP, i.e., event with magnitude greater than or equal to $M_0 - 0.5$ occurred in the 5 year period.

`TIP.level` is a number between -0.9 and 0.1, and is also referred to as the *critical series*, see $W(J_i)$ in "Details". Two consecutive positive values of `TIP.level` is equivalent to a TIP.

`g` is the number of different types of measures exceeding their critical value in the last 3 years.

`h` is the number of measures exceeding their critical value in the last 3 years.

Author(s)

Li Dongfeng, 1997; modified by Maaike Vreede, 1998.

See Also

`decluster.M8`, `M8`, `M8.series`

Examples

```
#   Requires the package ssNZ
require(ssNZ)

as.catalogue(subsetrect(NZ, minday=julian(1,1,1965),
                        maxday=julian(1,1,2000), minmag=4.5),
             catname="NZ45")

decluster.M8(NZ45, cutoff.mag=4.5, decluster.name="NZ.mainshocks")

temp1 <- M8.series(NZ.mainshocks, M0=7.0, minday=julian(y=1965, x=1, d=1),
                   start.series=julian(y=1975, x=1, d=1), centrelong=176,
                   centrelat=-39, end.training=julian(y=2000, x=1, d=1))

temp2 <- M8.TIP(temp1, M0=7.0, end.training=julian(y=2000, x=1, d=1))

print(cbind(time=temp1$series[,"time"], TIP=temp2$TIP,
            TIP.type=temp2$TIP.type,
            TIP.level=signif(temp2$TIP.level, digits=3)),
      quote=FALSE)
```

`plot.M8` *Plot M8 Series*

Description

Plot the seven M8 series plus the eighth critical series.

Usage

```
## S3 method for class 'M8':
plot(x, title="M8 Series and TIPs", ...)
```

Arguments

`x`	a list object produced by the function `M8.TIP`.
`title`	a title to be placed at the top of the matrix of eight graphs.
`...`	other options for formatting `"datetimes"` objects.

Details

This produces a 4×2 grid of graphs. The first seven are graphs of the seven series, and the eighth is the critical series. When the training mode is not `"moving"`, the 90th and 75th percentiles for the first 6 and 7th series are plotted, respectively. These percentiles represent the thresholds for each of the seven series. When the training mode is `"moving"` the threshold values are not constant over time, and are not plotted.

The plot of the "critical series" has a reference line marked at zero. A TIP is called if this series exceeds zero in two consecutive time periods. Points are marked on all plots where a TIP is in effect, the symbols having the following interpretations:

- Successful TIP (STIP): `pch = 5`
- Near Successful TIP (STIP-): `pch = 2`
- False TIP (FTIP): `pch = 4`
- Continuing TIP (CTIP): `pch = 3`
- TIP caused by quake of target magnitude (c.e.): `pch = 0`

Note that the plotting symbols may be different on different plotting devices, see the argument `pch` in the graphics function `par`.

Value

`NULL`

Author(s)

Li Dongfeng, 1997; modified by Maaike Vreede, 1998.

See Also

`M8`, `M8.series`, `M8.TIP`

Examples

```
#   Requires the package ssNZ
require(ssNZ)

as.catalogue(subsetrect(NZ, minday=julian(1,1,1965),
                        maxday=julian(1,1,2000), minmag=4.5),
             catname="NZ45")

decluster.M8(NZ45, cutoff.mag=4.5, decluster.name="NZ.mainshocks")

temp <- M8(NZ.mainshocks, M0=7.0, minday=julian(y=1965, x=1, d=1),
           start.series=julian(y=1975, x=1, d=1), centrelong=176,
           centrelat=-39, end.training=julian(y=2000, x=1, d=1),
           plotit=FALSE)

savpar <- par(no.readonly=TRUE)
plot(temp, title="M8 Analysis of NZ events")
par(savpar)
```

Index

∗Topic **distribution**
    emp.cdf, 5
∗Topic **documentation**
    Change Log, 2
∗Topic **hplot**
    plot.M8, 14
∗Topic **methods**
    plot.M8, 14
∗Topic **ts**
    decluster.M8, 4
    M8, 7
    M8.series, 8
    M8.TIP, 11

Change Log, 2
Changes, *3*
Changes *(Change Log)*, 2

decluster.M8, *2, 3*, 4, *7–10*, *13*

ecdf, *6*
emp.cdf, *3*, 5

in.circle, *2*

M8, *5*, 7, *10*, *13, 14*
M8.series, *2, 3*, *5*, *7*, 8, *8, 9*, *11–14*
M8.TIP, *2, 3*, *5–8*, *10*, 11, *12*, *14*

par, *14*
plot.M8, *2*, *5*, *7*, 14
prt.mainshocks, *2*

quantile, *3*, *5, 6*

stop, *3*
subsetrect, *3*
sumrow, *2*